AMENDMENTS TO THE CLAIMS

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

IN THE CLAIMS

Please cancel claims 1 through 15 without prejudice or disclaimer of the subject matter therein.

Please add new claims 16 through 46 as follows:

Claims 1-15 (cancelled)

16. (New) A method of mapping graphical block diagram block parameters in a graphical block diagram modeling environment, comprising:

      receiving a user-defined block parameter; and

      processing the user-defined block parameter to optimally produce a run-time block parameter.

17. (New) The method of claim 16, further comprising a block method inversely mapping the block run-time parameter to the user-defined block parameter to optimize block implementation.

18. (New) The method of claim 16, further comprising receiving a plurality of user-defined block parameters.

19. (New) The method of claim 18, further comprising processing the plurality of user-defined block parameters to optimally produce a run-time block parameter.

20. (New) The method of claim 19, wherein the plurality of user-defined block parameters is processed to produce a single run-time block parameter.

21. (New) The method of claim 19, wherein the run-time block parameter is configured to return at least one of simulation results, and automatically generated code that implements graphical block diagram model equations.

22. (New) The method of claim 19, further comprising mapping by discarding at least a portion of the plurality of user-defined block parameters to reduce memory requirements.

23. (New) The method of claim 19, further comprising pooling like non-interfaced run-time block parameters to reduce repetition of the non-interfaced run-time block parameters.

24. (New) The method of claim 23, wherein the pooling step comprises mapping user-defined block parameters into an existing pool.

25. (New) The method of claim 23, wherein the pooling step is repeated with additional optimization.

26. (New) The method of claim 19, further comprising mapping by translating the plurality of user-defined block parameters based at least in part on type.

27. (New) The method of claim 16, wherein the run-time block parameter is configured to return at least one of simulation results, and automatically generated code that implements graphical block diagram model equations.

28. (New) The method of claim 27, wherein when the code is automatically generated, the parameter expressions are maintained in the automatically generated code.

29. (New) The method of claim 28, wherein the parameter expressions contain interfaced variables that are updatable during modeling.

30. (New) The method of claim 29, further comprising converting to a relatively more compact representation portions of the parameter expressions that are constants while providing access to interfaced variables that are updatable.

31. (New) The method of claim 29, wherein interfaced variables are updatable.

32. (New) The method of claim 31, wherein updatable variables used in a plurality of blocks are declared only once.

33. (New) A method of mapping graphical block diagram block parameters in a graphical block diagram modeling environment, comprising:

receiving a plurality of user-defined block parameters;

processing the plurality of user-defined block parameter to optimally produce a plurality of run-time block parameters;

pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling.

34. (New) The method of claim 33, wherein pooling further comprises mapping user-defined block parameters into an existing pool.

35. (New) The method of claim 33, wherein the non-interfaced run-time block parameters have stored values that differ from presented values.

36. (New) The method of claim 35, wherein the non-interfaced run-time block parameters are fixed point.

37. (New) The method of claim 33, further comprising translating at run-time constant parameter values to an internal representation to enable increased pooling.

38. (New) The method of claim 33, wherein the step of pooling further comprises collecting constant portions of an expression containing an interfaced variable.

39. (New) The method of claim 33, wherein the run-time block parameters are configured to return at least one of simulation results, and automatically generated code that implements graphical block diagram model equations.

40. (New) The method of claim 39, wherein when the code is automatically generated, the parameter expressions are maintained in the automatically generated code.

41. (New) The method of claim 40, wherein the parameter expressions contain interfaced variables which are updatable.

42. (New) The method of claim 41, further comprising converting to a relatively more compact representation portions of the parameter expressions that are constants while providing access to interfaced variables.

43. (New) The method of claim 41, wherein interfaced variables are updatable.

44. (New) The method of claim 43, wherein updatable variables used in a plurality of blocks are declared only once.

45. (New) A medium for use in a graphical modeling environment on an electronic device, the medium holding instructions executable using the electronic device for performing a method of mapping graphical block diagram block parameters, the method comprising:

       receiving a user-defined block parameter; and

       processing the user-defined block parameter to optimally produce a run-time block parameter for use during modeling.

46. (New) A medium for use in a graphical modeling environment on an electronic device, the medium holding instructions executable using the electronic device for performing a method of mapping graphical block diagram block parameters, the method comprising:

receiving a plurality of user-defined block parameters;

processing the plurality of user-defined block parameter to optimally produce a plurality of run-time block parameters; and

pooling together like non-interfaced run-time block parameters to create a run-time parameter expression for use during modeling.